

SOAtest Smart API Test Generator

The SOAtest Smart API Test Generator enables you to capture traffic and create API tests from interactions with a web application. In this section:

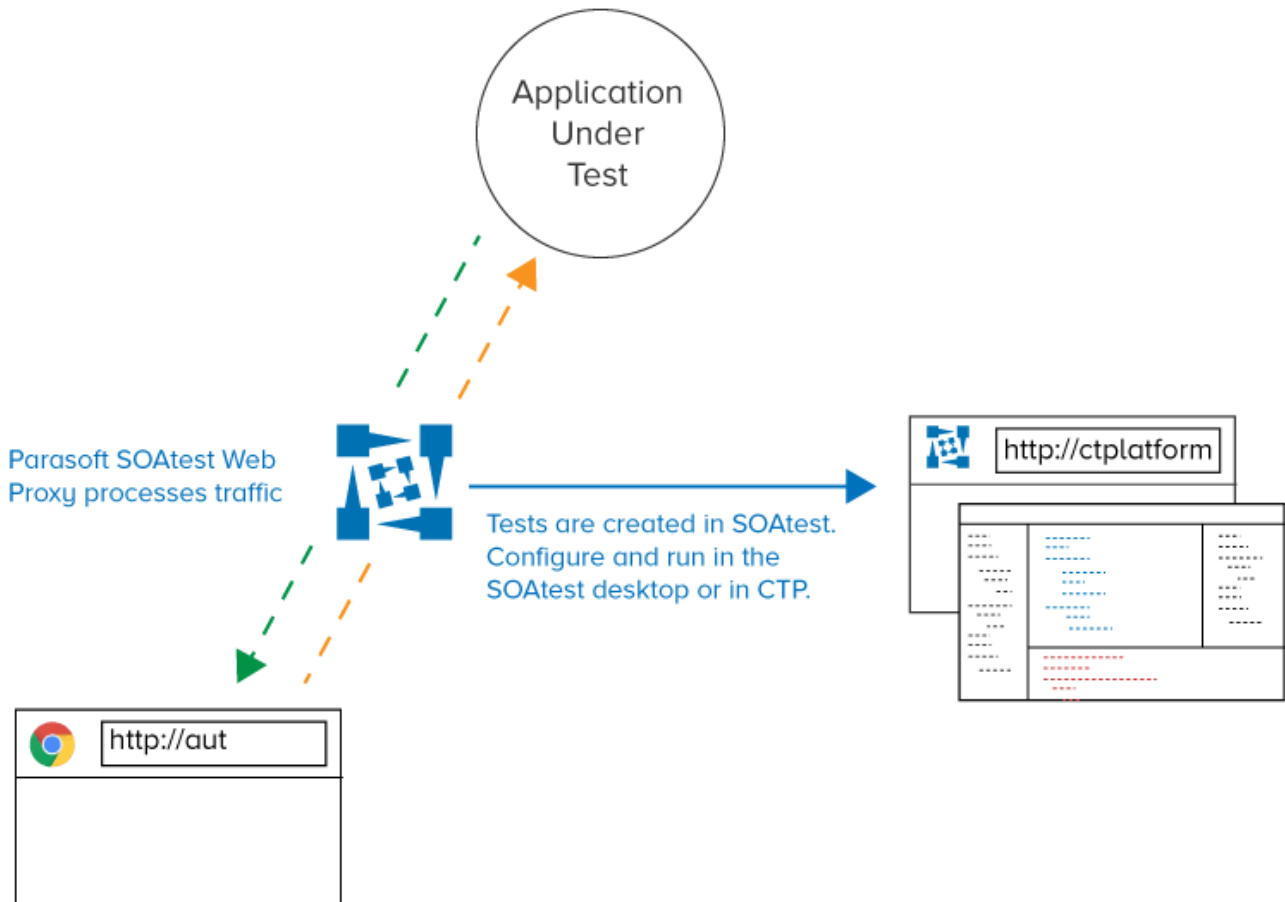
- Overview
- Requirements
- Licensing
- Installation
- Starting the Web Proxy
- Using the Browser Extension
- Advanced Configuration and Usage

Overview

The SOAtest Smart API Test Generator includes the following components:

- SOAtest Traffic Recorder: The traffic recorder is an extension for the Chrome browser. It is installed on the local machine and provides an interface for configuring your connection to SOAtest, as well as starting/stopping recording.
- Parasoft SOAtest Web Proxy: The proxy is installed on the local machine, processes traffic, and creates a .tst file in SOAtest.
- Parasoft Root Certificate Authority: The CA enables HTTPS traffic to be recorded.

The Smart API Test Generator connects to a SOAtest desktop or server, which enables you to configure and run the test.



Requirements

- SOAtest desktop or server 9.10.5 or later
- Latest version of Google Chrome

Licensing

You can freely download and install the Smart API Test Generator, but a valid license for the Advanced Test Generation license feature in SOAtest is required to record traffic and generate tests. The following license options are available:

- Advanced Test Generation Desktop: Enables you to create assets from the web proxy on the same machine where SOAtest is installed. Connections from remote machines are not allowed.
- Advanced Test Generation 5 Users: Up to five users can create assets from any machine that can connect to the SOAtest server.
- Advanced Test Generation 25 Users: Up to 25 users can create assets from any machine that can connect to the SOAtest server.
- Advanced Test Generation 100 Users: Up to 100 users can create assets from any machine that can connect to the SOAtest server.

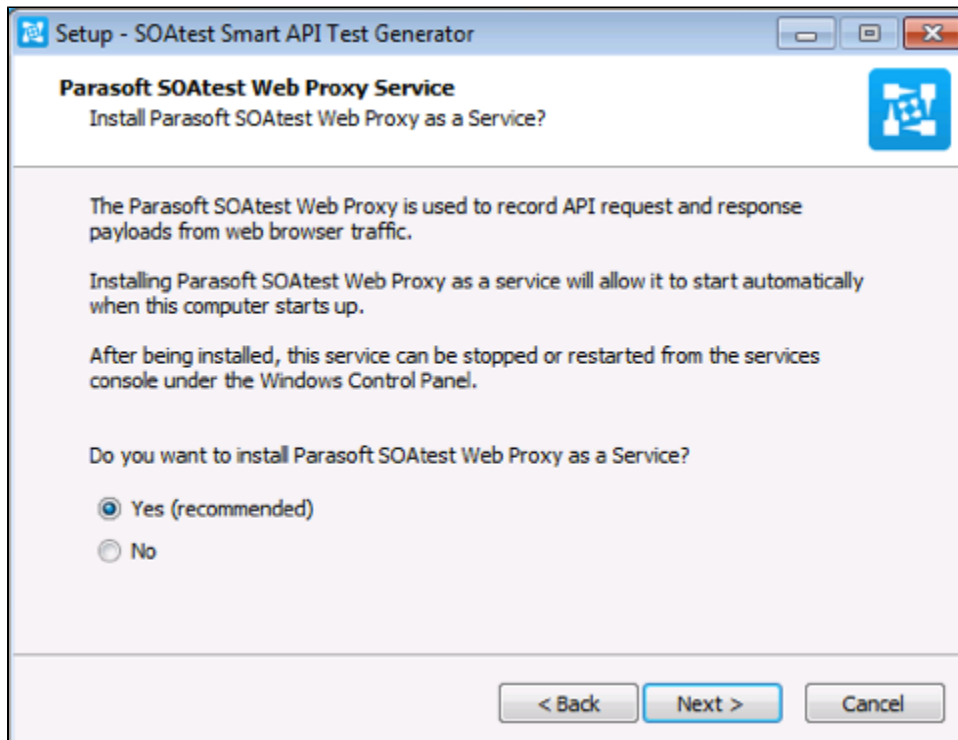
Contact your Parasoft representative for additional licensing information.

Installation

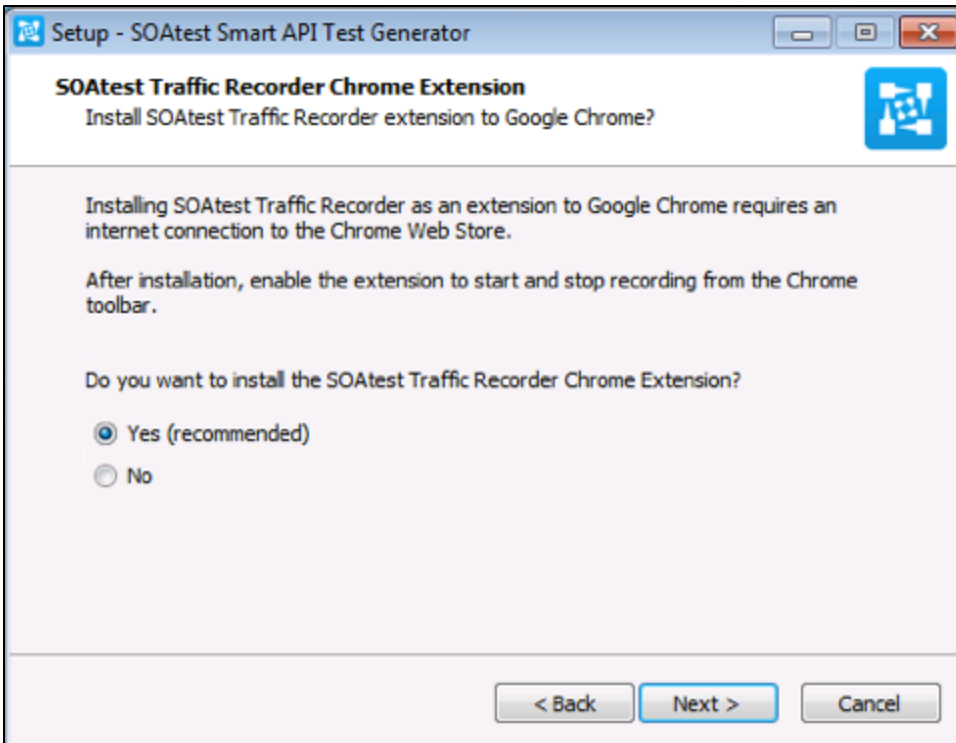
Run the `parasoft_soatest_smart_api_test_generator_<version>_win32.exe` file for installation on Windows or the `parasoft_soatest_smart_api_test_generator_<version>_macos.dmg` for installation on Mac OS. If Mac OS can not run the .dmg for security reasons, right-click the file and choose **Open With> Installer**.

Follow the wizard for your OS to complete the installation. See [Manual Installation](#) if you experience issues using the installer.

During Windows installation, you will be prompted to install the Parasoft SOAtest Web Proxy as a Windows service. This enables you to start and stop the proxy using the services console under the Windows Control Panel, as well as use the SOAtest Web Proxy console in the Windows Start menu (see [Starting the Web Proxy](#)). You can also manually start the proxy by running the executable JAR file (see [Manually Starting the Web Proxy](#)).

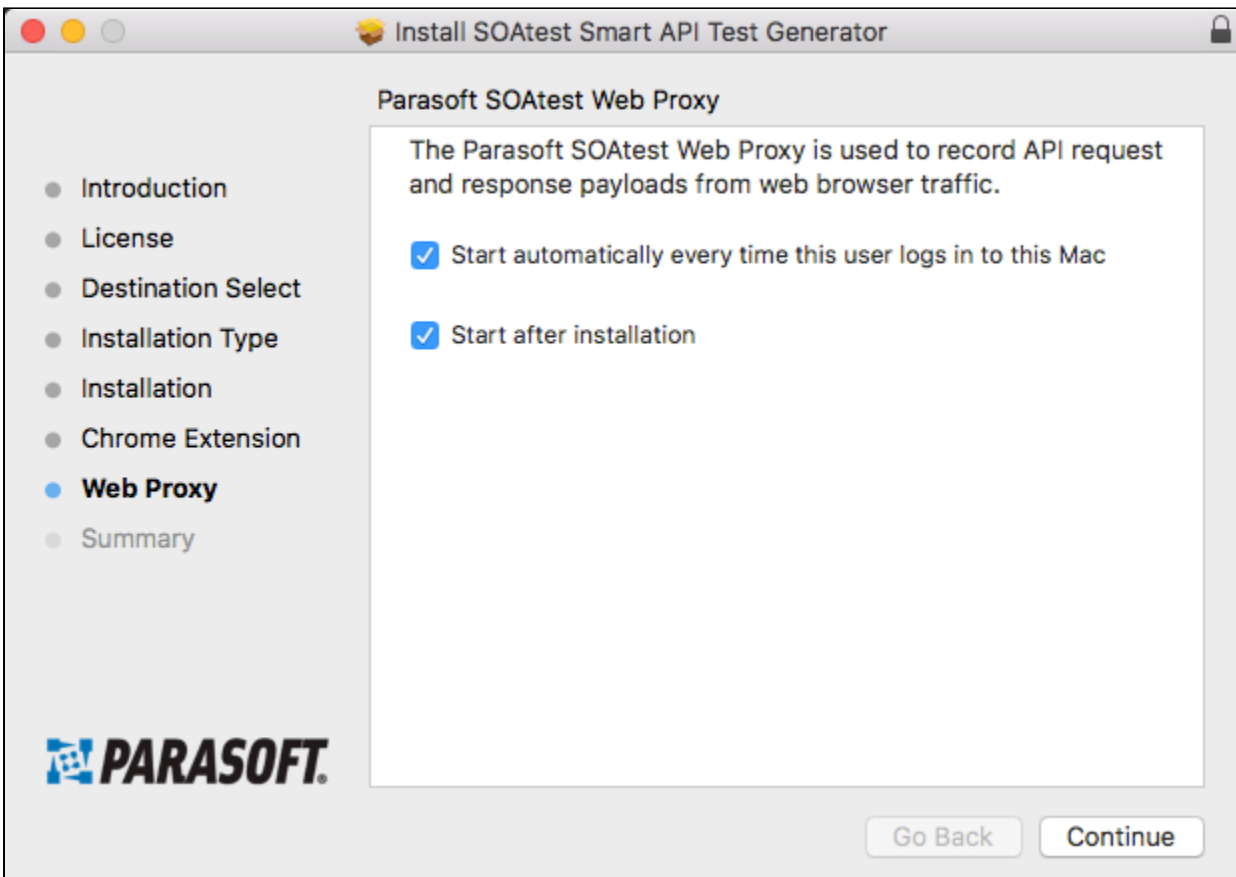


The Windows and Mac installer will also prompt you to install the SOAtest Traffic Recorder from the Google Chrome store (Windows shown). See [Manual Installation](#) if access to the Google Chrome store is prohibited or if you experience issues using the installer.

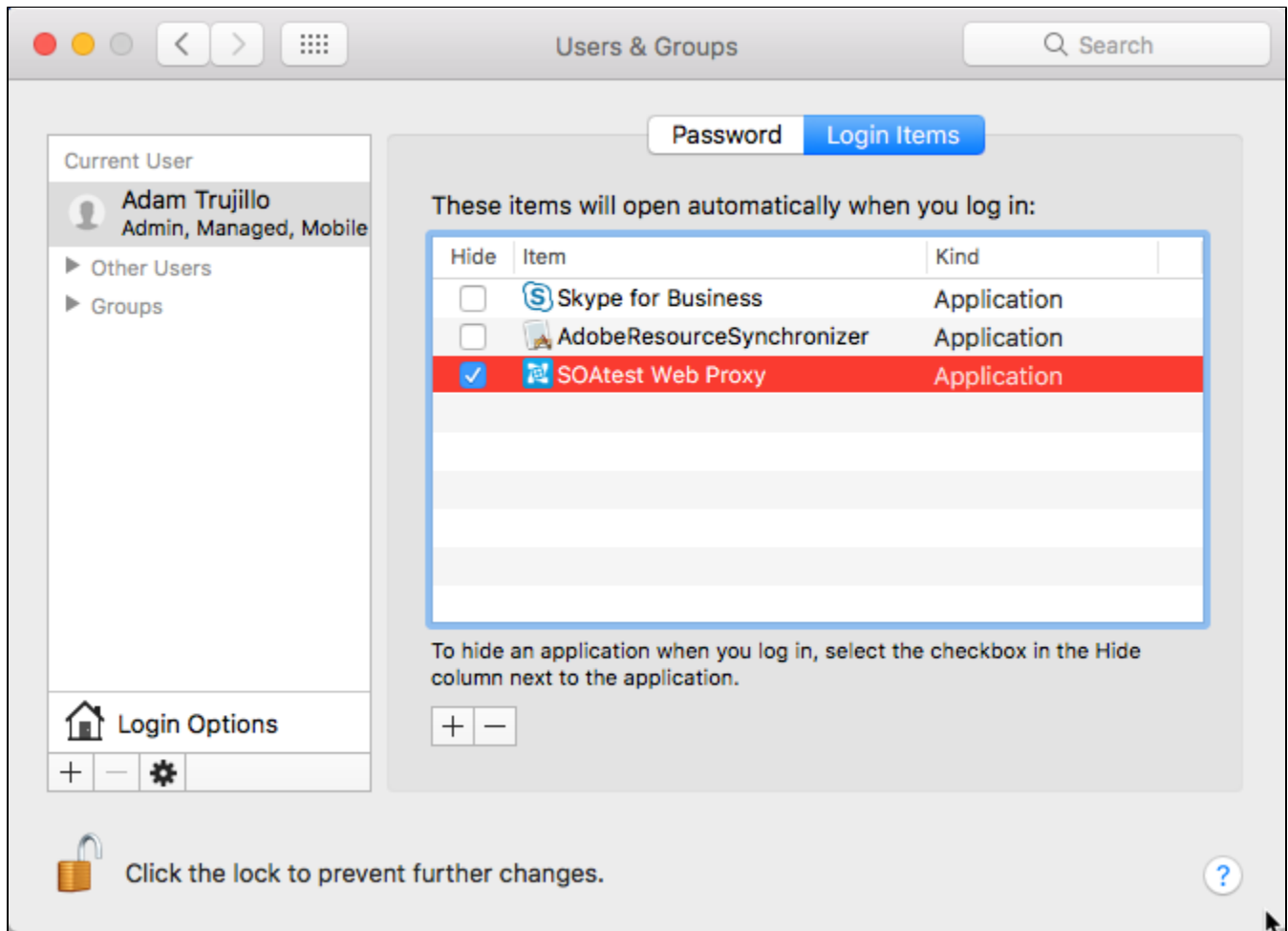


The installer will prompt you to install the Parasoft Root Certificate Authority if it is not detected. The CA is required to record traffic over HTTPS. The CA is normally installed along with the SOAtest/Virtualize desktop application. You can also add the CA manually (see [Installing the Parasoft Root Certificate Authority](#)).

During installation on Mac OS, you can enable the web proxy to automatically start whenever you log into the machine.

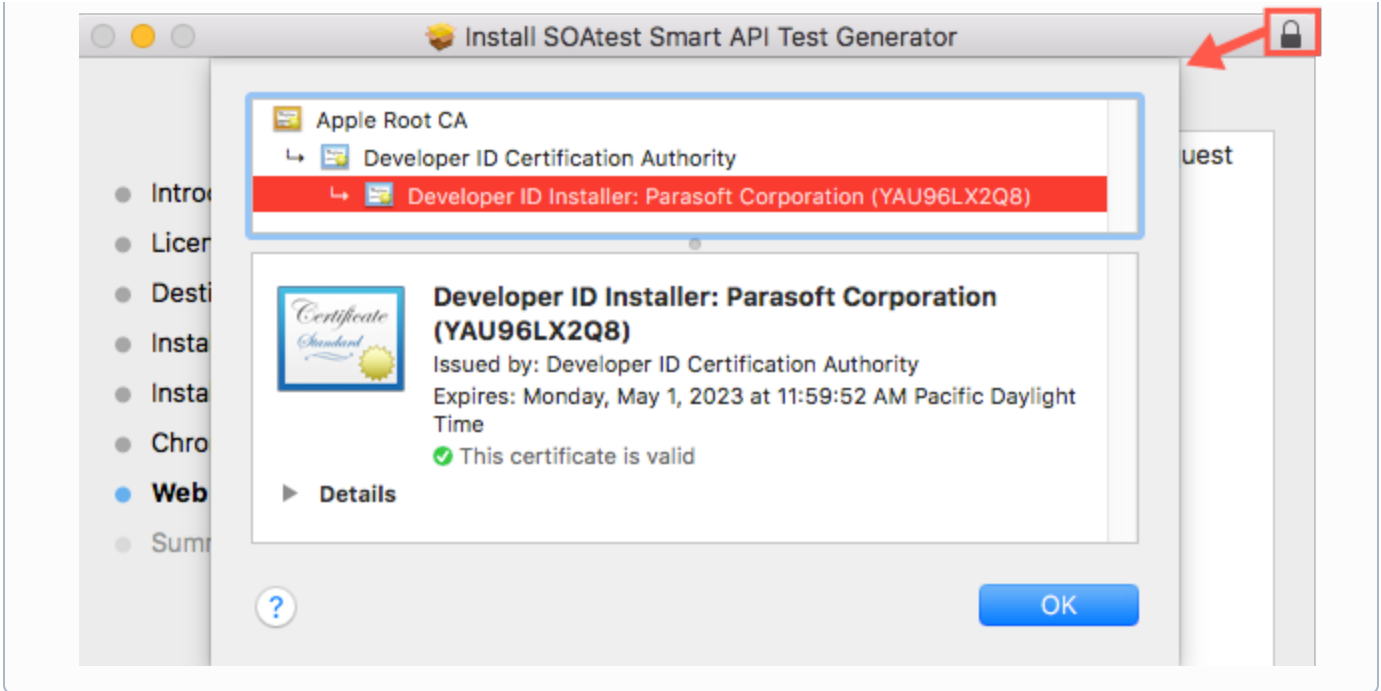


You can enable or disable automatic start up for the web proxy after installation by in the **System Preferences > Users & Groups > Login Items** settings.



Viewing the Developer ID Certificate Authority

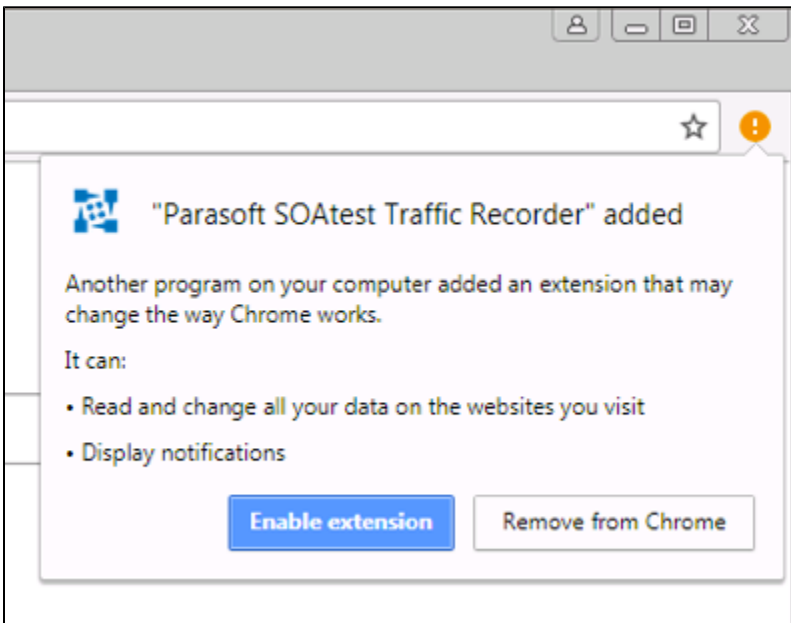
You can click the padlock icon to view the Apple developer ID certificate authority (CA). The certificate identifies Parasoft as a trusted source of software for Apple devices.



After the installer finishes, you will be prompted to enable the extension when you launch Chrome.

Do not click Remove from Chrome.

This action will uninstall and blacklist the extension. As a result, you will not be able to install or update the extension automatically.



The installer will prompt you to start the Parasoft SOAtest Web Proxy when the installation completes. If you choose to finish the installation without starting the proxy, see [Starting the Web Proxy](#) for instructions on starting the proxy on demand. If the proxy is running, see [Using the Browser Extension](#) for next steps. The SOAtest server must be running before you can use the extension.

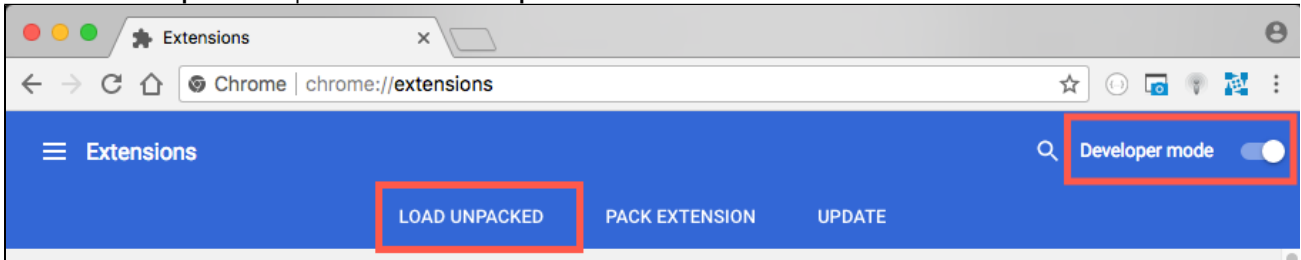
Uninstalling the SOAtest Smart API Test Generator

Use the regular Windows program uninstaller in the control panel to fully uninstall the tool. The Mac OS installer ships with an **Uninstall SOAtest Smart API Test Generator** utility that you can run if you wish to completely uninstall the tool. The Mac OS utility will be located in your Applications/Parasoft folder.

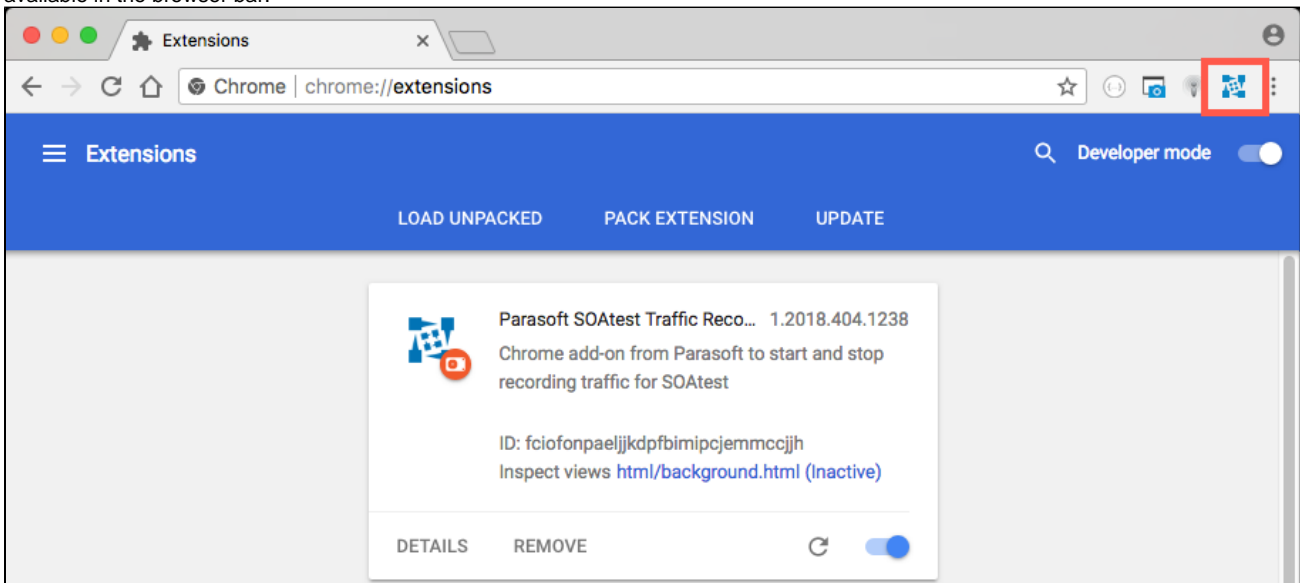
Manual Installation

The following instructions describe how to manually install the SOAtest Smart API Test Generator components if you are unable to use the Windows installer.

1. Download the TrafficRecordingExtension.zip folder (contact your Parasoft representative).
2. Extract the contents of the compressed folder anywhere on your machine. No additional installation is necessary to run the Parasoft SOAtest Web Proxy.
3. Open a Chrome browser window and open the Extensions page (chrome://extensions).
4. Enable the **Developer mode** option and click **Load unpacked extension**.



5. Navigate to and choose the [TrafficRecordingExtension_INSTALL]/chrome-extension directory to install the extension. The tool will be available in the browser bar.



The Parasoft SOAtest Web Proxy and SOAtest server must be running before you can use the extension (see [Starting the Web Proxy](#)). If the SOAtest server is running and the default configuration is used to start the proxy, you will be able to start using the extension to record traffic. See [Using the Browser Extension](#) if these components are running. You will also need to configure connections to the Parasoft SOAtest Web Proxy and SOAtest server once the browser extension is installed.

Installing the Parasoft Root Certificate Authority

The Parasoft Root Certificate Authority must be installed as a trusted certificate to enable recording of HTTPS traffic. If the certificate authority is not installed, the browser will not consider the connection secure and may refuse to allow the connection.

Windows

The Parasoft Root Certificate Authority is normally installed automatically by both the SOAtest Windows installer and the SOAtest Smart API Test Generator installer. You can also run the soatestcli with the `-installcertificate` option to manually install it if necessary. For example:

```
soatestcli -installcertificate
```

If SOAtest is not installed on the local machine, you can manually add the certificate to Chrome:

1. Open the Chrome settings page (chrome://settings) and expand the **Advanced** section.
2. Click **Manage certificates** in the Privacy and security section.
3. Click the **Trusted Root Certification Authorities** tab in the Certificates dialog.
4. Click **Import** to click Next when the Certificate Import Wizard opens.

5. Browse to the [TrafficRecordingExtension_INSTALL]/parasoft.cer file and click **Next**.
6. Verify that the wizard will place the certificate in the Trusted Root Certification Authorities folder and click **Next**.
7. Click **Finish** and confirm that you want to add the Parasoftware Root Certificate Authority if prompted.

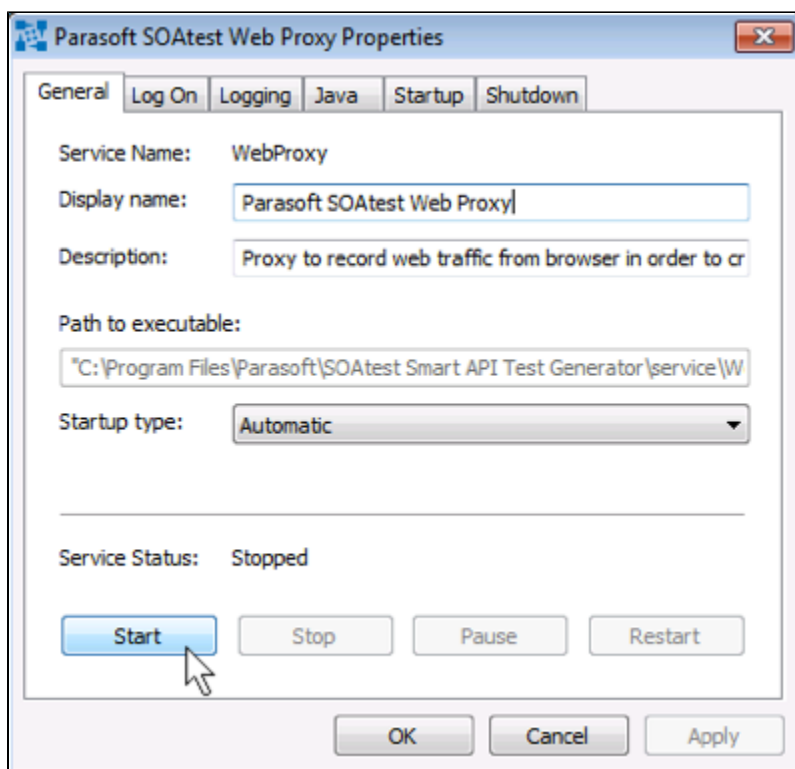
Mac OS

1. Open the Chrome settings page (chrome://settings) and expand the **Advanced** section.
2. Click **Manage certificates** in the Privacy and security section. The Keychain Access application will open.
3. Choose **login** under Keychains and choose **Certificates** under Category.
4. Choose **File> Import items** and navigate to parasoft.cer in the [TrafficRecordingExtension_INSTALL] directory.
5. Right-click the Parasoftware Root Certificate Authority when it appears and choose **Get Info**.
6. Expand the Trust section and choose **Always Trust** from the **When using this certificate** drop-down menu.
7. Enter your credentials when prompted to save the configuration.

Starting the Web Proxy

If you installed the Parasoftware Web Proxy as a Windows service, you can use the SOAtest Web Proxy console to start and stop the proxy.

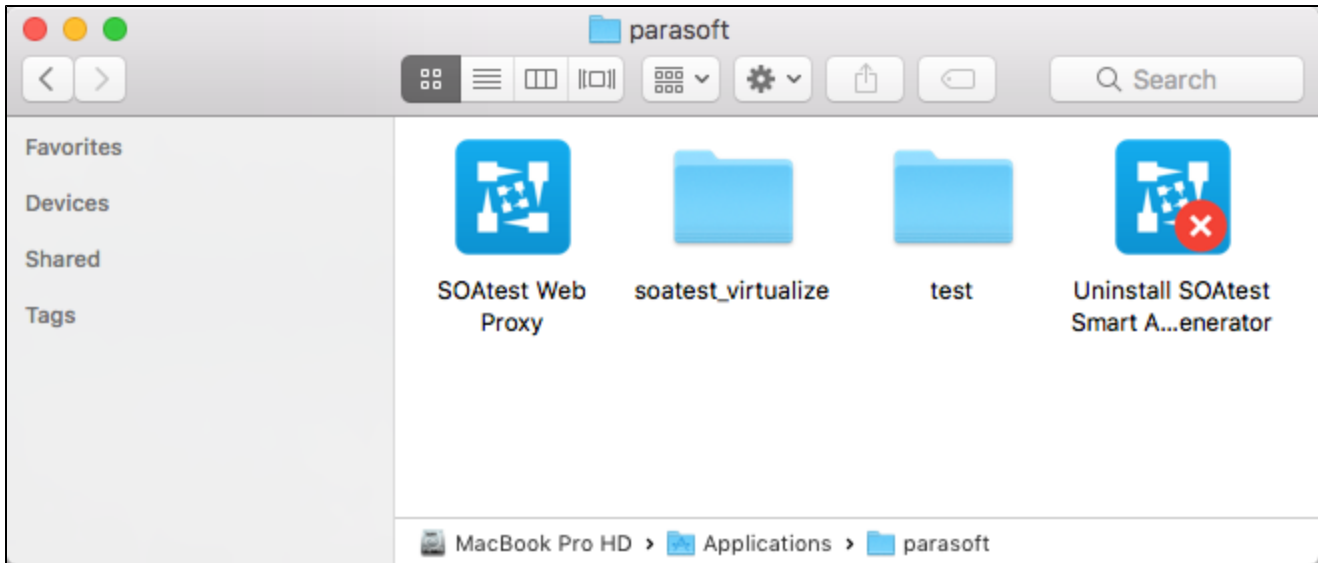
1. Choose **SOAtest Web Proxy** from the Windows Start menu or double-click the desktop icon (if you chose this option during installation).
2. Click **Start** in the General tab.



The web proxy API will be started on port 40090 by default. The proxy port used for recording will be assigned by scanning for open ports starting with port 40443. You can configure port numbers and other options when starting the proxy in the web proxy console. See [Proxy System Properties](#) for additional information.

Open the console and click **Stop** to shutdown the proxy.

On Mac OS, the web proxy will be located in your Applications/Parasoftware folder.



Manually Starting the Web Proxy

The web proxy is a JAR file that you can run with the `java` command:

```
%INSTALLDIR%/jre/bin/java -jar com.parasoft.traffic.proxy.jar
```

Java Support

The Smart API Test Generator ships with its own instance of Java, but you can use any Java 8 or later that is already installed on your system.

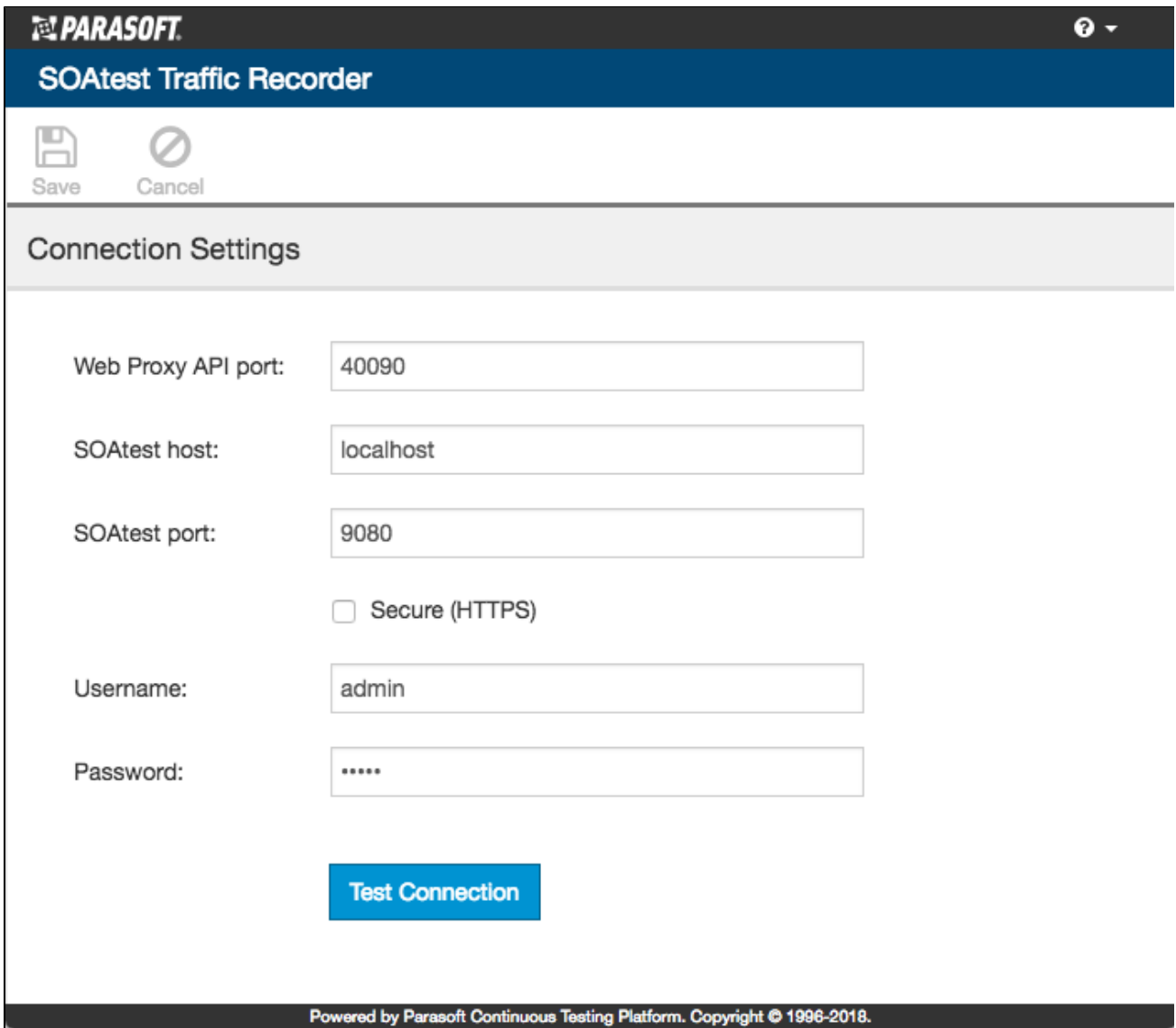
The web proxy API will be started on port 40090 by default. The proxy port used for recording will be assigned by scanning for open ports starting with port 40443. You can specify system properties to configure port numbers and other options when starting the proxy. See [Proxy System Properties](#) for additional information.

Using the Browser Extension

The browser extension provide the interface for configuring connections to the SOAtest server and the web proxy, starting and stopping traffic recording, and saving the traffic as `.tst` files.

Configuration

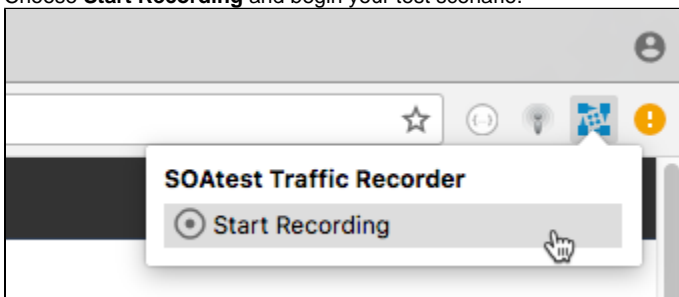
1. Right-click the Parasoft icon and choose **Options** to open the configuration settings. You can also access the configuration settings by clicking the **Options** link on the `chrome://extensions` homepage.
2. Specify the proxy API port (see [Proxy System Properties](#)) and SOAtest server connection settings.



- Enable the **Secure (HTTPS)** option to communicate with SOAtest over HTTPS.
- Click **Test Connection** to verify your settings and click **Save** if successful.

Usage

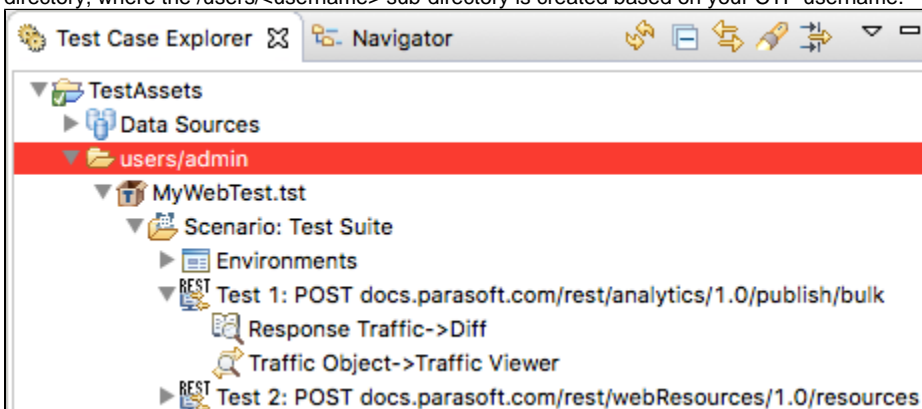
- Navigate to the web application you want to generate tests from and click the Parasoft icon.
- Choose **Start Recording** and begin your test scenario.



- When finished, click the logo and choose **Stop Recording**.

- Specify information for the test when prompted. Test names must be unique. You can also associate tests with artifacts in your requirements and/or defect tracking system.

- Click **Create Test Asset** when ready or **Discard Recording** to exit the test creation window without generating the test file.
- When the SOAtest Traffic Recorder finishes processing the recorded traffic, you can review the test file in the TestAssets project of the SOAtest Test Case Explorer or in the API Testing module in CTP. Parasoft .tst files will appear under the TestAssets/users/<username> directory, where the /users/<username> sub-directory is created based on your CTP username.



If you are using the Smart API Test Generator with a local instance of SOAtest, your tests will appear in the TestAssets/users/anonymous directory.

Advanced Configuration and Usage

You can configure the proxy port, specify test creation options, and access the API so that the tool can be used in automated workflows.

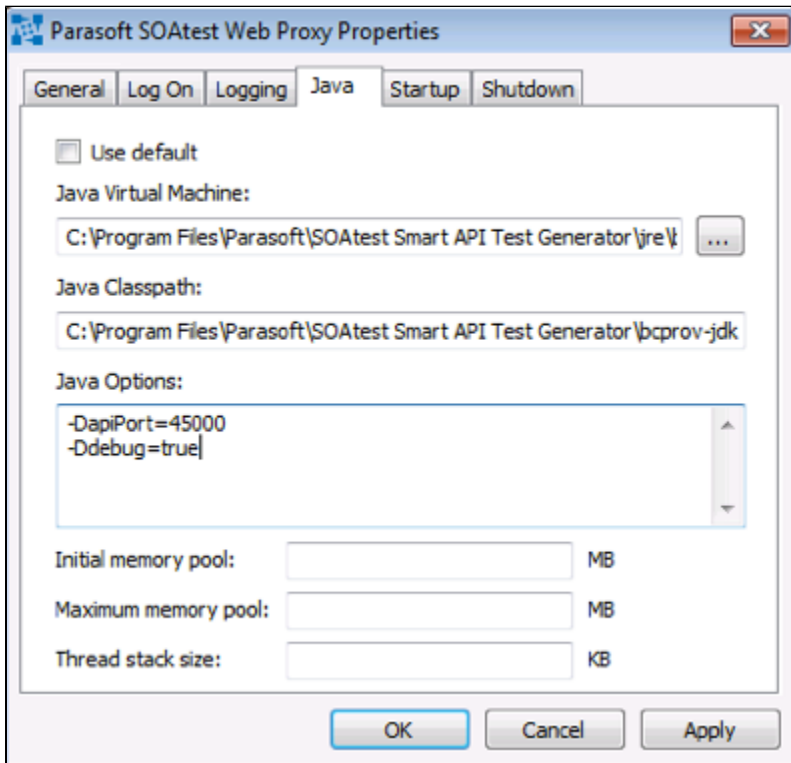
Proxy System Properties

You can configure the web proxy by specifying system properties in the Manage Web Proxy console or from the command line when executing the .jar file. You can specify the following properties:

-DapiPort	System property that specifies the API port number. Default is 40090.
-DproxyPortStart	System property that specifies the starting port number when scanning for an open port. Default is 40443.
-DproxyPortRange	System property that limits the number of ports scanned after the starting port number (value specified with the <code>proxyPortStart</code> argument). Set to 0 to prevent scanning.
-DuseSystemProxy	System property that enables using the network-level proxy. Default is <code>true</code> .
-Ddebug	Enables/disables system debugging information. Default is <code>false</code> .
-Duser.home	<p>(Optional) Specifies a location for writing traffic files. A <code>/parasoft/recorded_traffic</code> subdirectory will be created in the specified location.</p> <p>The default location depends on how the JAR is started and system architecture.</p> <p>If the JAR is started manually on a Windows x86/x64 system, the default will be <code>%USERPROFILE%\parasoft\recorded_traffic</code>.</p> <p>If the JAR is started as a Windows service or through the SOAtest Web Proxy utility, the default will be <code>%windir%\System32\config\systemprofile\parasoft\recorded_traffic</code> on Windows x86 and <code>%windir%\SysWOW64\config\systemprofile\parasoft\recorded_traffic</code> on Windows x64.</p>

Manage Web Proxy Example

Add the system properties to the Java Options field in the Java tab. The following example sets the API port to 45000 and enables debugging:



Command Line Example

Specify the configuration options on startup from the command line. The following startup command sets the API port to 10090, scans for a range of port numbers, and enables debugging:

```
%INSTALLDIR%/jre/bin/java -DapiPort=10080 -DproxyPortStart=20443 -
DproxyPortRange=10 -Ddebug=true -jar com.parasoft.traffic.proxy.jar
```

Java Support

The Smart API Test Generator ships with its own instance of Java, but you can use any Java 8 or later that is already installed on your system.

Test Creation Properties

You can determine how the recorder generates tests by defining properties in the [TrafficRecordingExtension_INSTALL]/tst_creation.properties configuration file. This file should be in the same directory as the com.parasoft.traffic.proxy.jar file so that it can automatically be read during test creation. You can specify the following settings:

disableDiffCreation	Enables/disables diff creation. See Diff for additional information. Default is <code>false</code> .
disableDataBankCreation	Enables/disables data bank creation. See Data Exchange Tools for additional information. Default is <code>false</code> .
disableEnvironmentCreation	Enables/disables the creation of an environment and environment variables. Default is <code>false</code> .
includeContentTypes	Defines a comma-separated list of content types to include during traffic processing. Default is <code>application/json</code> .
excludeContentTypes	Defines a comma-separated list of content types to exclude during traffic processing. Default is empty.
disableDiffParameterization	Enables/disables diff parameterization. Parameterization enables you to use data banked values in the diff. If diff parameterization is disabled (setting this property to <code>true</code>), only static values will be available. Default is <code>false</code> .
diffToolIgnoreNames.<number>	<p>Defines a regex matching element names that should be ignored when creating diffs. Default is <code>(?i)^(time date url href).*</code></p> <p>You can specify additional name patterns by adding properties and appending them with <code>.<number></code>.</p> <p>For example:</p> <pre>diffToolIgnoreNames.1=<name_pattern_1> diffToolIgnoreNames.2=<name_pattern_3> diffToolIgnoreNames.3=<name_pattern_3></pre>
diffToolIgnoreValues.<number>	<p>Defines a regex matching values that should be ignored when creating diffs. Default is to ignore timestamps:</p> <pre>[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}(\.[0-9]{1,3})?((\+ -)[0-9]{2}:[0-9]{2}) Z)?</pre> <p>You can specify additional value patterns by adding properties and appending them with <code>.<number></code>.</p> <p>For example:</p> <pre>diffToolIgnoreValues.1=<value_pattern_1> diffToolIgnoreValues.2=<value_pattern_3> diffToolIgnoreValues.3=<value_pattern_3></pre>

Example tst_creation.properties File

```
includeContentTypes=application/json
excludeContentTypes=image/png,font/ttf,text/css,application/javascript
disableDiffCreation=false
disableDiffParameterization=false
disableEnvironmentCreation=false
disableDataBankCreation=false
# Ignore values that start with "time", "date", "url" or "href" in diff
tool, case insensitive
diffToolIgnoreNames.1=(?i)^(time|date|url|href).*
# Ignore values that end with "time", "date", "url" or "href" in diff
tool, case insensitive
diffToolIgnoreNames.2=(?i).*(time|date|url|href)$
# Ignore values like a timestamp in diff tool, e.g. 2018-03-21T07:00:00.000
Z
diffToolIgnoreValues.1=[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]
{2}([\.[0-9]{1,3})?(([\+-][0-9]{2}:[0-9]{2})|Z)?
```

Using the API

After starting the web proxy, you can go to the following URL to review the API definitions:

<http://<host>:<port>/api/openapi.json>